

E-Commerce Sales & Customer Feedback Analysis (SQL project)

Mihirkumar Chaudhary



Business Problems

1.How does customer feedback influence product sales?

-Using SQL queries to aggregate sales data and identify **top-performing products** based on **total revenue** and **units sold**.

3.How can feedback data improve marketing decisions?

-Using CASE statements to categorize products based on average ratings and sales performance, then recommending marketing strategies based on the analysis.

2.Which products generate the highest revenue and volume?

-Using SQL queries to aggregate sales data and identify top-performing products based on total revenue and units sold.

4.Who are the most valuable customers for retention strategies?

-Querying purchase frequency and total spent per customer, then classifying them into high-value or low-value segments based on total spend and customer feedback ratings.

Objectives

01

Design a normalized relational database for e-commerce data

02

Perform SQL-based sales and feedback analysis

03

Create reusable views and stored procedures

04

Translate insights into business recommendations

Tech Stack

1.SQL (Joins, Aggregations,
CASE, Statements)

2.Relational Database
Diagram (ER modeling)

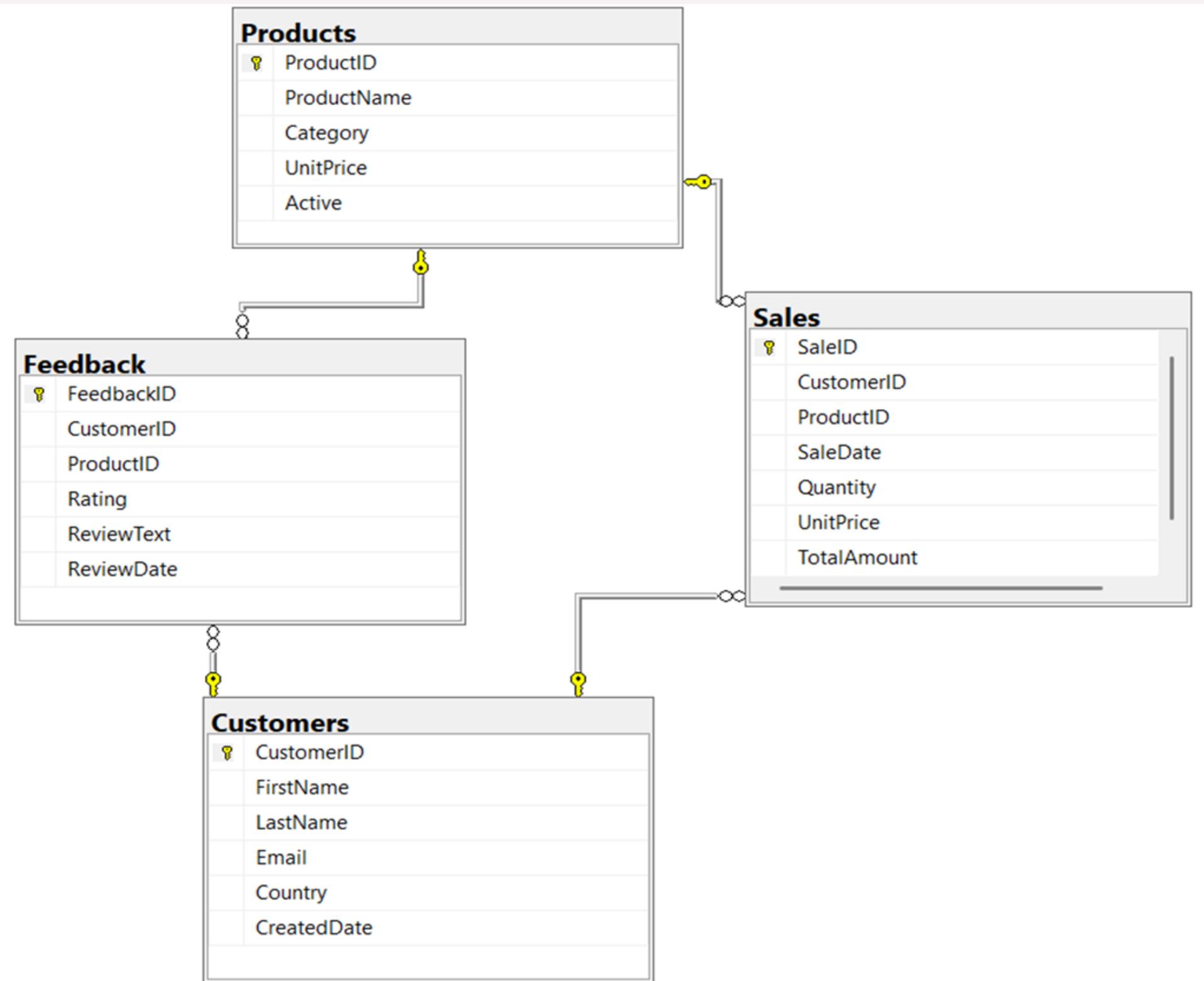
3.Views & Stored
Procedures

4.Customer
Segmentation

5. Sales & Feedback
Analytics

6. Business Insight
Generation

ER Diagram



Customer Purchase Summary

```
SELECT
    c.CustomerID,
    c.FirstName + ' ' + c.LastName AS CustomerName,
    COUNT(s.SaleID) AS TotalOrders,
    SUM(s.TotalAmount) AS TotalSpent
FROM Customers c
LEFT JOIN Sales s ON c.CustomerID =
    s.CustomerID
GROUP BY c.CustomerID, c.FirstName,
    c.LastName
ORDER BY TotalSpent DESC;
```

	CustomerID	CustomerName	TotalOrders	TotalSpent
1	5	Bobby Patel	2	144.98
2	1	Aisha Patel	2	99.97
3	3	Vaibhav Jaiswal	2	57.49
4	4	Vishal Thakor	1	35.00
5	2	Siddhi Shah	1	25.00

- Calculates total orders and total spending per customer
- Identifies high-value customers for loyalty programs
- Supports targeted promotions and reflection strategies

Product Rating Analysis

```
SELECT
    p.ProductName,
    AVG(CAST(f.Rating AS
DECIMAL(10,2))) AS AvgRating,
    COUNT(f.FeedbackID) AS
TotalReviews
FROM Feedback f
JOIN Products p ON f.ProductID =
p.ProductID
GROUP BY p.ProductName
ORDER BY AvgRating DESC;
```

	ProductName	AvgRating	TotalReviews
1	Desk Lamp	5.000000	1
2	Gaming Keyboard	4.500000	2
3	Wireless Mouse	4.500000	2
4	Yoga Mat	3.500000	2
5	Coffee Mug	2.000000	1

- Analyzes average customer ratings for products
- Identifies high-rated products for promotional focus
- Highlights low-rated products needing improvement

Product Performance Classification

SELECT

```
p.ProductName,  
AVG(CAST(f.Rating AS DECIMAL(10,2))) AS  
AvgRating,  
CASE  
    WHEN AVG(CAST(f.Rating AS DECIMAL(10,2)))  
>= 4.5 THEN 'Excellent'  
    WHEN AVG(CAST(f.Rating AS DECIMAL(10,2)))  
>= 3.5 THEN 'Good'  
    WHEN AVG(CAST(f.Rating AS DECIMAL(10,2)))  
>= 2.5 THEN 'Average'  
    ELSE 'Poor'  
END AS RatingCategory  
FROM Products p  
LEFT JOIN Feedback f ON p.ProductID =  
f.ProductID  
GROUP BY p.ProductName;
```

	ProductName	AvgRating	RatingCategory
1	Coffee Mug	2.000000	Poor
2	Desk Lamp	5.000000	Excellent
3	Gaming Keyboard	4.500000	Excellent
4	Wireless Mouse	4.500000	Excellent
5	Yoga Mat	3.500000	Good

- Used CASE statements to classify products as Excellent, Good, Poor
- Converts raw feedback into actionable performance categories
- Enables data-driven marketing and product strategy

Sales Trend Analysis

```
SELECT  
    SaleDate,  
    SUM(TotalAmount)  
        AS DailyRevenue  
    FROM Sales  
    GROUP BY SaleDate  
    ORDER BY SaleDate;
```

	SaleDate	DailyRevenue
1	2024-11-01	39.98
2	2024-11-03	59.99
3	2024-11-05	25.00
4	2024-11-06	19.99
5	2024-11-07	72.50
6	2024-11-08	119.98
7	2024-11-10	25.00

- Tracks revenue trends over time
- Identifies peak sales periods
- Supports demand forecasting and inventory planning

Top Selling Product Analysis

```
SELECT TOP 5
    p.ProductName,
    SUM(s.Quantity) AS UnitsSold,
    SUM(s.TotalAmount) AS
    Revenue
FROM Sales s
JOIN Products p ON s.ProductID
    = p.ProductID
GROUP BY p.ProductName
ORDER BY Revenue DESC;
```

	ProductName	UnitsSold	Revenue
1	Gaming Keyboard	3	179.97
2	Wireless Mouse	3	59.97
3	Yoga Mat	2	50.00
4	Coffee Mug	3	37.50
5	Desk Lamp	1	35.00

- Identifies top 5 Products by revenue and units sold
- Highlights customer preferences
- helps prioritize inventory and promotions

Automated Product Performance Reporting

```
CREATE PROCEDURE
usp_GetProductPerformance
    @ProductID INT
AS
BEGIN
    SELECT
        p.ProductName,
        SUM(s.TotalAmount) AS TotalRevenue,
        SUM(s.Quantity) AS UnitsSold,
        AVG(CAST(f.Rating AS DECIMAL(10,2))) AS
        AvgRating
    FROM Products p
    LEFT JOIN Sales s ON p.ProductID =
    s.ProductID
    LEFT JOIN Feedback f ON p.ProductID =
    f.ProductID
    WHERE p.ProductID = @ProductID
    GROUP BY p.ProductName;
END;
```

	ProductName	TotalRevenue	UnitsSold	AvgRating
1	Coffee Mug	37.50	3	2.000000
2	Desk Lamp	35.00	1	5.000000
3	Gaming Keyboard	359.94	6	4.500000
4	Wireless Mouse	119.94	6	4.500000
5	Yoga Mat	100.00	4	3.500000

- Built stored procedure to fetch revenue, units sold, and ratings per product
- Demonstrates real-world reporting automation
- Enables scalable analytics for dashboards

Sales KPI Automation

```
CREATE PROCEDURE  
usp_GetSalesKPI  
@StartDate DATE,  
@EndDate DATE,  
@TotalRevenue DECIMAL(18,2)  
OUTPUT  
AS  
BEGIN  
SELECT  
@TotalRevenue =  
SUM(TotalAmount)  
FROM Sales  
WHERE SaleDate BETWEEN  
@StartDate AND @EndDate;  
END;
```

Commands completed successfully.

TotalRevenue	
1	362.44

- Uses OUTPUT parameters to return revenue KPIs
- Useful for dashboards and management reporting
- Demonstrates advanced SQL capabilities

Conclusion

Summarize Key Findings

- Identified top-performing products and revenue drivers
- Discovered products with strong ratings but low sales potential
- Segmented customers based on value and engagement

Business Insights

- Supports data-driven marketing decisions
- Improves customer retention strategies
- Enhances reporting efficiency using SQL automation

Thank You

This project demonstrates how SQL and analytics can transform raw data into meaningful business insights.

Open to feedback, collaboration, and analytics opportunities.